|  |
| --- |
| claudiusmarius |
| GUIDE DE REALISATION DE LA CARTE DE DEVELOPPEMENT POUR ATTINY85 – NEW CONCEPT |
| https://www.youtube.com/c/ClaudeDufourmont |
|  |
| **Claude DUFOURMONT** |
| **28/05/2023** |



|  |  |  |
| --- | --- | --- |
| Indice | Date | Observations |
| 0 | 28/05/23 | Création du document |
|  |  |  |
|  |  |  |

|  |
| --- |
| J’ai voulu faire ce guide afin que chacune ou chacun s’intéressant à l’ATtiny85, puisse réaliser cette carte de développement pour ce mini et très intéressant MCU, certes qui peut être programmé à l’aide d’un Arduino et des fils volants de manière ponctuelle, mais avec beaucoup de manipulations durant la phase de développement et avec des risques d’erreur, tandis que cette plaquette peu onéreuse, très pratique vous aidera et vous incitera à utiliser vos ATtiny85 beaucoup plus souvent |

Table des matières

[1 > Préambule-Avertissements 2](#_Toc136362631)

[2 > Présentation du schéma 2](#_Toc136362632)

[3 > Présentation de la plaquette 4](#_Toc136362633)

[4 > Disposition des composants 6](#_Toc136362634)

[5 > Programmation de l’Arduino UNO en tant que programmateur ISP 8](#_Toc136362635)

[6 > Téléversement d’un code dans l’ATtiny85 8](#_Toc136362636)

[7 > Liste des composants 11](#_Toc136362637)

[8 > Conseils, Retour d’expérience 12](#_Toc136362638)

[8.1. Coupure des pattes de composants traversant : 12](#_Toc136362639)

[8.2. Axe d’amélioration sur le schéma : 12](#_Toc136362640)

[8.3. Nettoyage PCB : 12](#_Toc136362641)

[Annexe 1 – Extrait datasheet DG333 13](#_Toc136362642)

[Annexe 2 – Schéma KiCad 14](#_Toc136362643)

# 1 > Préambule-Avertissements

Je ne suis qu’un maker électronicien à la retraite ayant du temps à consacrer à sa passion. J’essaie de faire profiter les personnes pouvant être intéressées par mes idées, mes réalisations, BENEVOLEMENT, je suis complètement désintéressé par l’argent enfin je veux dire que je déconnecte complétement mes partages avec l’idée de faire de l’argent, donc tout lien, toute information pouvant être assimilé à l’idée de monnayer ma passion et de m’enrichir en faisant des bénéfices est faux, je partage juste mes informations sans contrepartie. Par contre je n’apprécierais pas que certains utilisent mes informations pour les monnayer sans accord écrit de ma part.

Sur l’aspect sécurité, chacun est responsable de ses propres actions et des conséquences potetielles. Je me dégage de toute responsabilité.

# 2 > Présentation du schéma

Nous utiliserons un Arduino Nano pour servir de programmateur ISP. Ses pinoches dédiées SPI seront utilisées

|  |  |  |
| --- | --- | --- |
|  | PinOutNANO | Schéma |
| D12 | MISO | NanoMISO |
| D11 | MOSI | NanoMOSI |
| D10 | SS | NanoRST |
| D13 | SCK | NanoSCK |

Ces 4 ports sont envoyés à l’ATtiny85 via le commutateur analogique (analog switch) U2 (DG333 de chez VISHAY) durant la phase de téléversement (IN = HIGH pinoche D3), l’ATtiny85 (U3) est donc déconnecté totalement de la carte Aduino Nano. Ensuite lorsque la phase de téléversement sera terminée, le système s’en apercevra et passera IN à LOW et l’aiguillage de U2 sera inversé et isolera totalement l’ATtiny85 de la carte Arduino Nano en le reliant au montage via un clone U4 qui est en fait une prolongation le permettant d’être déporté sur un breadboard.

Hormis lors de la préparation de la carte Arduino Nano en tant que programmateur le commutateur J3 aura son cavalier sélectionné sur les pinoches 2 et 3 de manière à positionner C1 (10µf) sur RST de la carte Arduino Nano

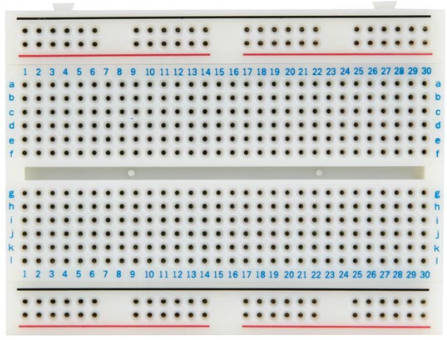
Vous retrouverez un extrait du datasheet de l’analog switch DG333 en annexe1.

Comme il a été dit un peu plus haut U2 orientera automatiquement l’ATtiny85 soit vers l’Arduino Nano (D3 = HIGH) pendant le téléversement soit vers l’extension (U4) lorsque l’Arduino Nano n’est pas ou n’est plus en phase de téléversement (D3 = LOW). Une adaptation du logiciel Arduino ISP a été faite dans une version que je vous propose, rendant cette commutation automatique. La LED D4 (bleue) permet de connaître dans quelle phase nous nous trouvons.

**ATTENTION IMPERATIF: Pour U2 surtout ne pas utiliser un MAX333 qui ressemble beaucoup au DG333, mais dont la RSon est beaucoup, beaucoup plus élevée que ce dernier.**

Les résistances de limitation de courant dans les LEDs D1,D2,D4,D5 ont été calculées grâce à mon « LED calculator » (vidéo YouTube : **DFT\_**[**#A48**](https://www.youtube.com/hashtag/a48)**LED CALCULATOR - NEW CONCEPT - NEO CONCEPT POUR DETERMINER LA RESISTANCE SERIE D'UNE LED**). Ce système permet via un petit montage connecté à l’IDE Arduino de visionner sur le sérial monitor la valeur de résistance à mettre en série (selon la tension d’alimentation choisie) pour avoir le degré de luminosité souhaité. Ne soyez donc pas étonnés de la dispersion des valeurs de résistances pour ces 4 LEDs de couleur différente.

Les 4 connecteurs J1, J2, J7, J8 n’ont pas de rôle fonctionnel proprement dit (elles ne sont reliées à rien électriquement), leur rôle est de pouvoir enficher le PCB sur une breadboard Velleman SDAD102 si vous choisissez cette solution (celle que je préfère, j’ai dimensionné le PCB en conséquence)



Les 4 PADS MH1 à MH4 ont été représentés sur le schéma pour pouvoir les utiliser en tant que 4 PADs VIA sur le PCB pour servir de trous de fixation, en les reliant à la masse sur le schéma, cela permet de les relier au plan de masse du PCB.

Les 2 quadruples pinheaders J4 et J5 permettent de disposer de pinoches d’alimentation 5V sur le PCB, attention tenir compte de ce qu’est capable de fournir la carte Arduino Nano (courant assez faible), sur la breadboard pour des montages consomment significativement je préconise un 5V dédié, en reliant les 2 GND bien entendu (GND PCB avec GND de l’alimentation 5V dédié).

Je rappelle qu’en extrémité droite de la carte à l’emplacement U4, il faut placer en face inférieure 2 fois 4 pinheaders males afin de pouvoir simuler la présence d’un ATtiny85 sur une breadboard, ici, une attention particulière est nécessaire sur la position verticale de ces pinoches pour favoriser au mieux « l’enfichabilité » de ces pinoches sur la breadboard.

J6 permet d’alimenter en externe le PCB (à ne surtout pas confondre avec le 5V dédié) via un barrel jack relié au Vin de l’Arduino nano de manière à pouvoir tester le montage hors USB (qu’il sera préférable de débrancher dans cette phase), la tension typiquement devra être compris entre 7 et 12V par exemple par une pile « agrafe » 9V 6LR61.

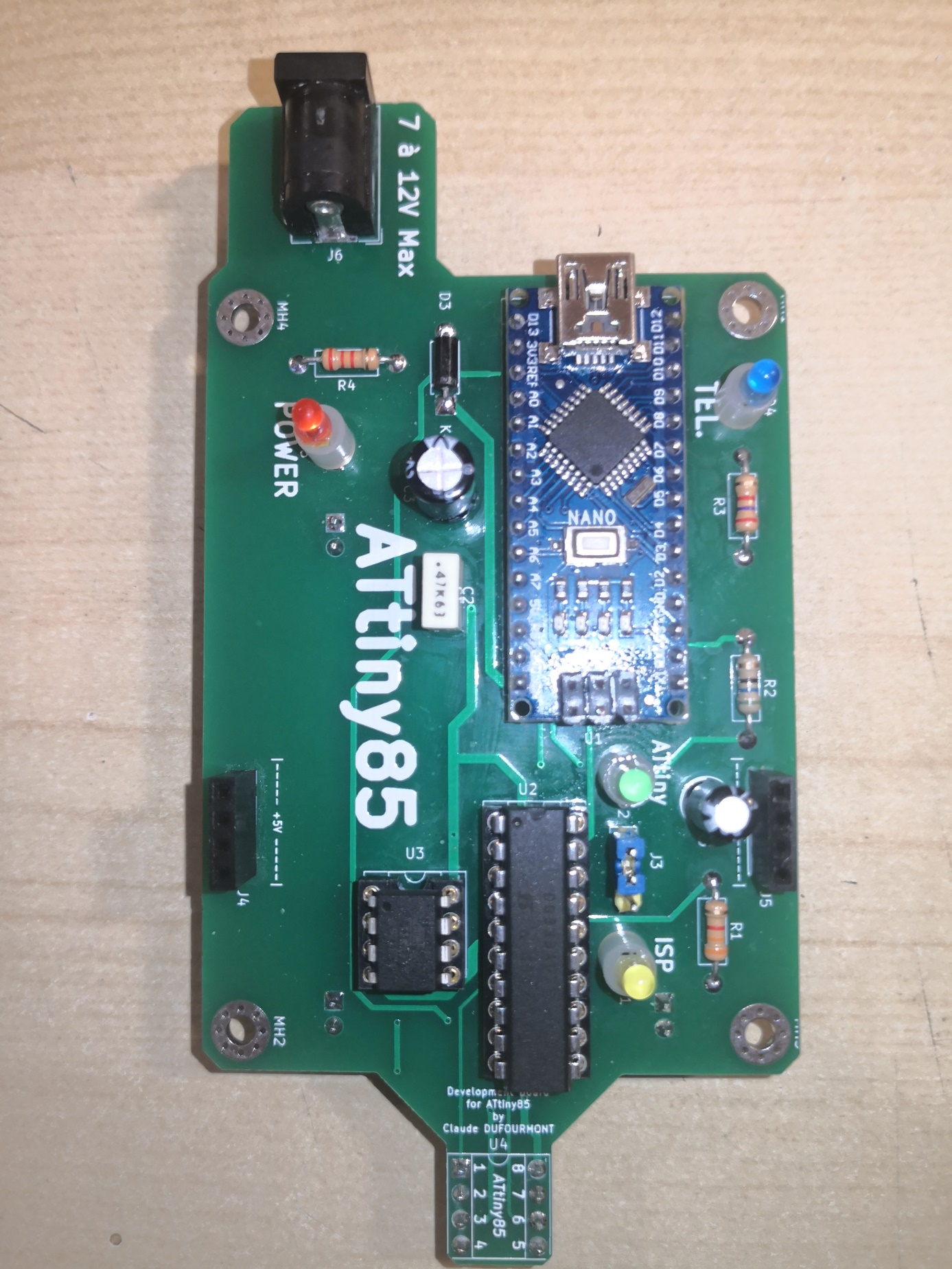
L’Arduino nano pourra être monté soit directement sur le PCB (soudé) soit si vous préférez sur 2 pinheaders femelles 15 broches. Les pads ont été optimisés pour ces 2 solutions.

Le schéma se trouve en annexe 2.

# 3 > Présentation de la plaquette

Plaquettes vierges face supérieure et face inférieure





**Carte Arduino Nano alimentée**

**Clone ATtiny85 à poser sur breadboard**

**Bornier +5V Arduino Nano**

**Bornier GND Arduino Nano**

**Cavalier J3 en position pour préparer la carte Nano en programmateur Arduino ISP**

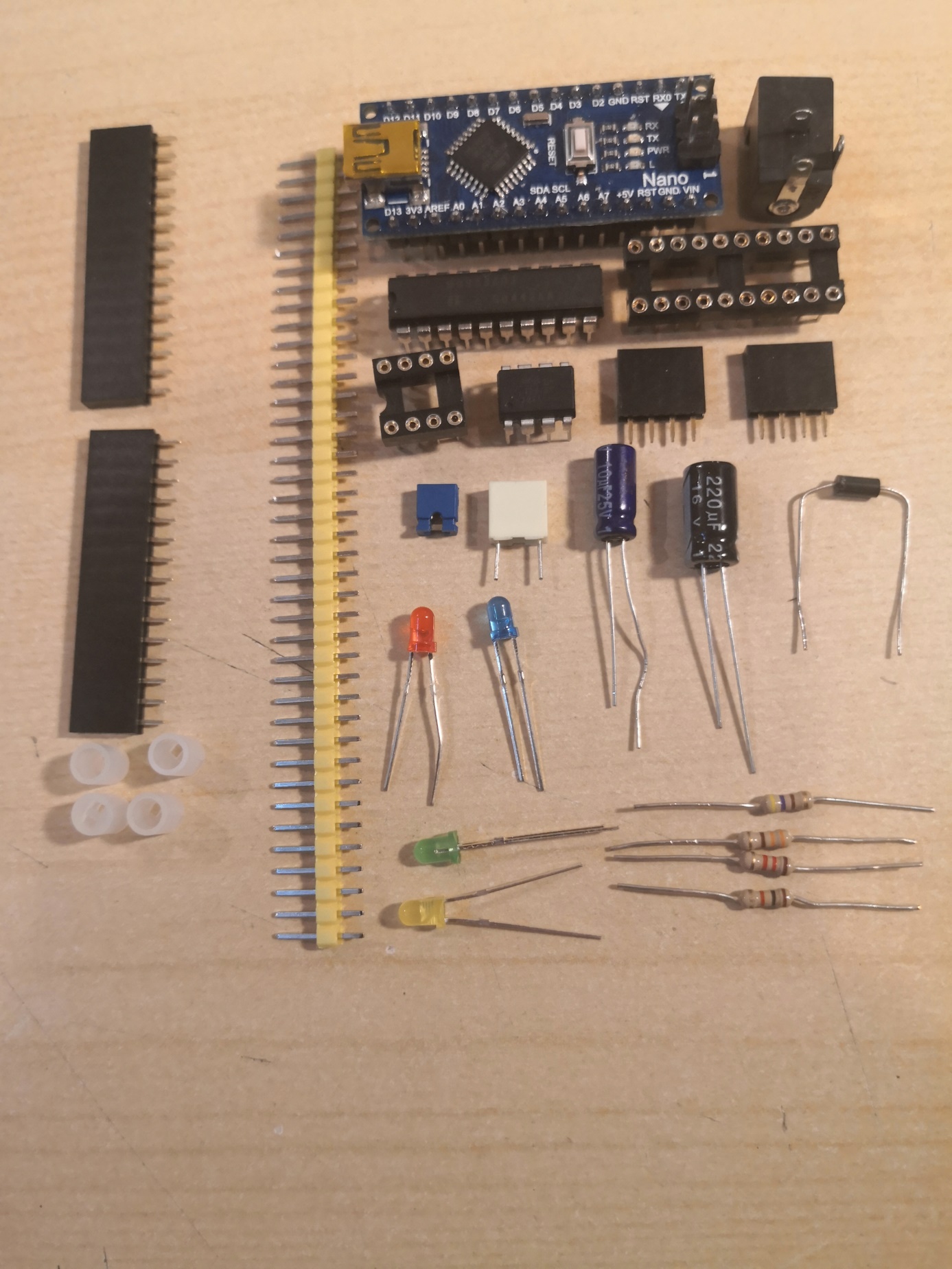
**Cavalier J3 en position pour téléversement code ATtiny85**

**Téléversement code ATtiny85 en cours**

**Connexion USB**

**Alimentation externe**

# 4 > Disposition des composants



**NOTA :** Ne lisez pas les valeurs des résistances (regardez le schéma plutôt), j’ai choisi 4 résistances au hasard pour la photo

Avant le soudage je contrôle visuellement mon PCB vierge et je teste mes LEDs, si possible sur breadboard avec les résistances qui leur sont prédestinées.

Je fais les contrôles électriques (absence de courts circuits, absence de coupure) sur mon PCB vierge.

Je contrôle ma carte Arduino Nano surtout si elle est prévue d’être soudée directement sur le PCB.

Pour le soudage des composants j’utilise un simple fer à souder campingaz (électrique contrairement à ce que le nom de la marque pourrait faire penser) 30/15W sur position 30W, pointe fine, et de la soudure 10/10mm.

Ensuite je fixe les supports du DG333 (U2) et de l’ATtiny85 (U3), puis tout le reste des composants (sauf la carte Arduino Nano) selon ma préférence (si possible terminer par les LEDs surtout si on leur laisse des longues pattes.

Ne pas oublier que les 4 pinheaders J1, J2, J7, J8 sont fixés en face inférieure (donc les soudures sont en face supérieure), attention couper assez court en face supérieure les pinoches de J1, surtout si vous avez prévu de souder la carte Arduino Nano, afin de supprimer tout risque d’affleurement de celles ci avec la face inférieure de la carte Arduino Nano.

Ne pas oublier non plus (mais c’est moins grave) de fixer en face inférieure les 2 fois 4 pinheaders de U4 (donc les soudures sont en face supérieure).

Pour les composants polarisés : **ATTENTION AU SENS**.

Nota : pour le barrel jack bien que les trous de passage aient été optimisés, il existe encore un certain jeu pouvant favoriser un positionnement désaxé du barrel, ne pas hésiter à poser une légère pointe de colle au moment de sa pose afin que celle-ci soit parfaite et débuter la soudure du barrel.

Couper les pattes au fur et à mesure. (Voir chapitre 8 Conseils et retour d’expérience)

Une fois tout cela réalisé et donc avant la pose de la carte Arduino Nano testée, je nettoie mon PCB à l’aide d’alcool isopropylique pour enlever tout le flux de soudure : bien rincer le circuit, frotter à la l’aide d’une vieille brosse à dents, rincer, peaufiner au coton tige.

Refaire un contrôle visuel de l’ensemble.

Poser la carte Arduino nano (**ATTENTION AU SENS**), ne souder dans un premier temps que les pinoches utiles (au cas où vous auriez à devoir la redéposer).

Faire des essais sous tension sans l’ATtiny85 et sans le DG333, vérifier tout ce qui peut être vérifiable : distribution des tensions, allumage des LEDs.

Terminer les soudures de la carte Arduino Nano et finaliser le nettoyage de la carte à l’alcool isopropylique.

Poser l’ATtiny85 et le DG333 sur leur support respectif (**ATTENTION AU SENS**).

Votre carte est prête, bonne chance vous pouvez la tester en réel.

# 5 > Programmation de l’Arduino UNO en tant que programmateur ISP

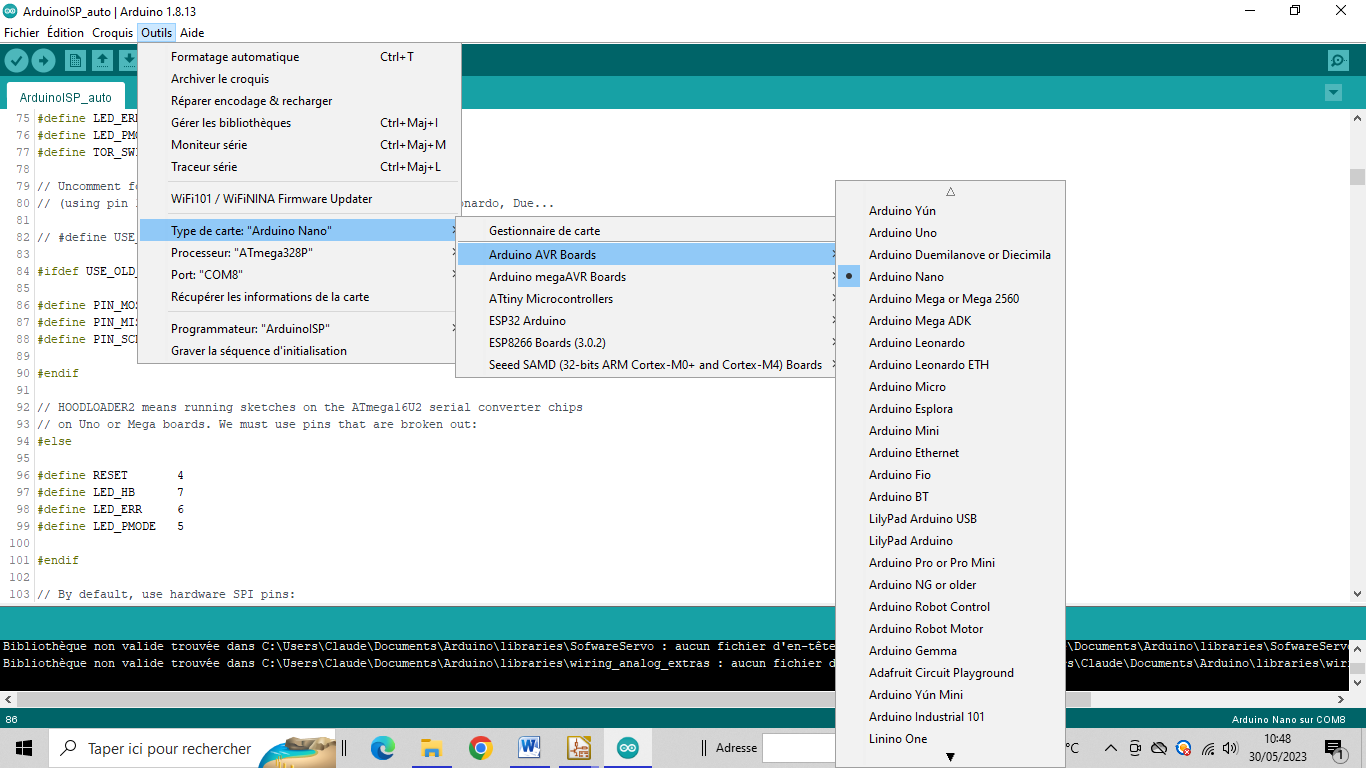
Cette étape consiste à injecter dans la carte Arduino Nano un code permettant de la transformer en programmateur ISP, elle sera sans doute à refaire à chaque fois que vous replacerez une carte Arduino Nano si vous avez opté pour une l’option carte montée sur pinheaders.

Si vous avez opté pour l’option carte soudée, cette étape est à faire une fois pour toutes.

**IMPERATIF :** Pour cette option le condensateur C1 ne doit pas être connecté au RST de la carte Arduino Nano. Il faut donc que le cavalier de J3 soit placé à droite, la LED jaune D1 s’allumera signifiant le bon positionnement dès que vous aurez connecté la carte Arduino Nano à l’IDE via votre PC.

Vous devrez téléverser le code que je vous fournis : **ArduinoISP\_auto** et non pas celui de l’IDE car celui-ci n’est pas prévu pour un fonctionnement automatique.

Dans outils de l’IDE opérez cette sélection :



**IMPORTANT :** vérifiez bien, dans la colonne de gauche que c’est bien l’option « ArduinoISP » qui est sélectionnée et la cohérence de sélection du port USB.

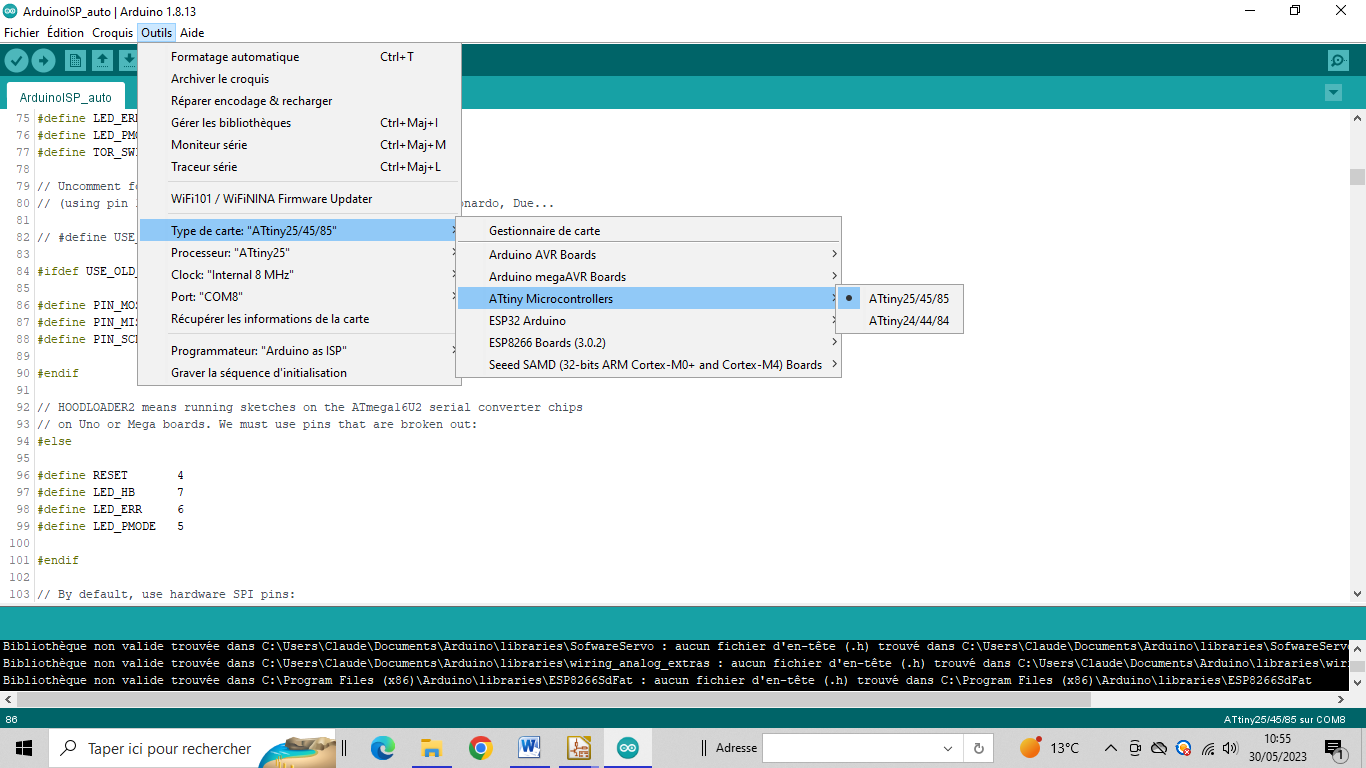
Lancez le téléversement. Une fois le téléversement terminé, placez le cavalier de J3 en position gauche, la LED verte D2 (ATtiny) s’allume.

Vous allez pouvoir passer à la programmation de votre ATtiny85.

# 6 > Téléversement d’un code dans l’ATtiny85

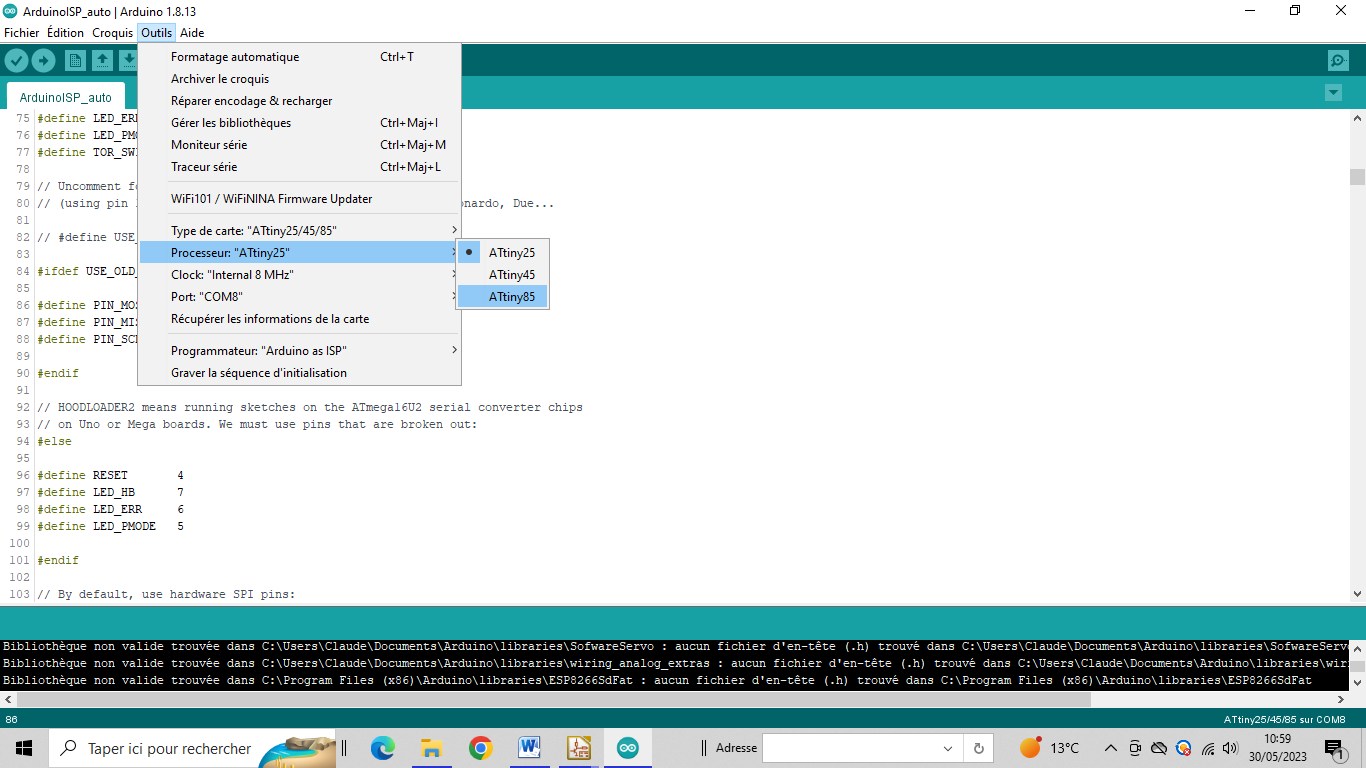
En préalable vous devrez installer les cartes ATtiny sur votre IDE, il y a tout un tas de tutos sur le sujet je vous conseille celui-ci : <https://www.youtube.com/watch?v=FhWHOeNo7FU&t=8s> à partir du timing 2 : 28

Gardez votre cordon USB branché et opérez ce choix dans l’IDE :

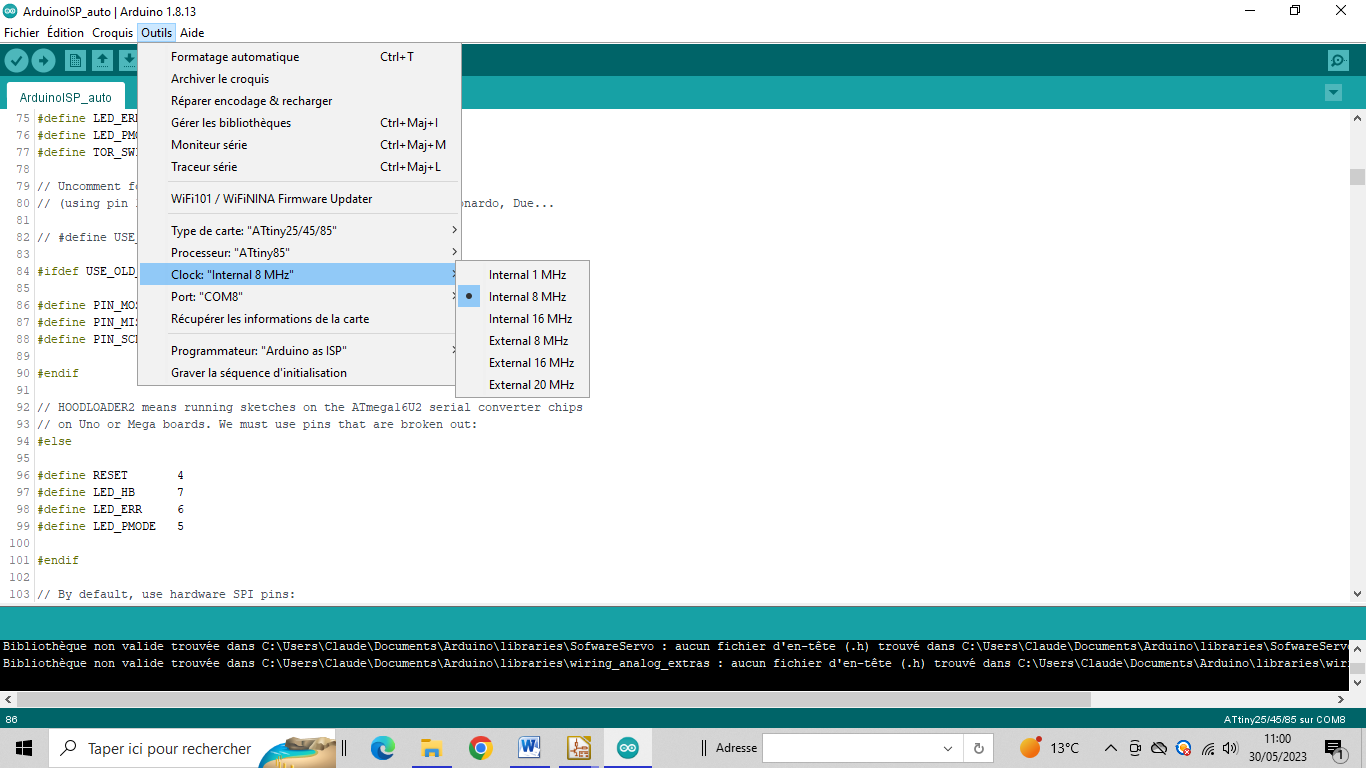


Veillez à ce que dans la colonne de gauche Programmateur « Arduino as ISP » est sélectionné (ne pas confondre avec le choix précédent « ArduinoISP »

Sélectionnez ensuite le choix d’ATtiny comme ceci :



**ATTENTION :** Ensuite vous sélectionnez le choix suivant pour la fréquence :

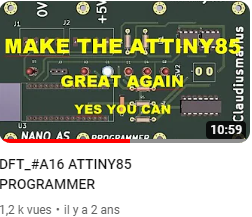


**INTERNAL et non pas EXTERNAL très, très important.**

Ensuite vous gravez la séquence (important pour valider ce choix de fréquence). Tant que votre ATtiny restera le même vous n’aurez en principe plus la nécessité de le faire.

Ensuite votre plaquette étant toujours reliée par USB et U4 posé sur breadboard avec le circuit correspondant à votre essai en place, vous pouvez injecter votre petit code d’essai. Lorsque vous verrez le début du téléversement apparaissant dans l’IDE, vous devrez constater de la LED bleue D4 (TEL.) s’allume et lorsqu’elle s‘éteindra votre ATtiny85 sera raccordé au montage implanté sur votre breadboard prêt à fonctionner vous pourrez ensuite ajuster le code à volonté sans devoir déplacer l’ATtiny85.

Je recommande de toujours laisser le même ATtiny85 en place sur la plaquette. Lorsque vous aurez une fois la phase de développement terminée, de programmer un autre avec le même programme bien sûr et en utilisant mon programmateur ATtiny 85 et 84 :



Voici le lien vidéo si vous préférez : [**https://youtu.be/1XOWKyjqMR4**](https://youtu.be/1XOWKyjqMR4)

# 7 > Liste des composants

|  |  |  |  |
| --- | --- | --- | --- |
| Repère | Valeur | |  |
| R1 | 3,3 | K | J'achète mes résistances chez Gotronic code article : 04099 |
| R2 | 55 | Ohms |  |
| R3 | 2,7 | K |  |
| R4 | 2,2 | K |  |
| D1 | LED | Jaune | 3mm |
| D2 | LED | Verte |  |
| D3 |  |  | 1N4148 |
| D4 | LED | Bleue |  |
| D5 | LED | Rouge |  |
| C1 | 10 | µf | 50V |
| C2 | 470 | nf |  |
| C3 | 220 | µf | 16V |
| U1 |  |  | Arduino Nano |
| U2 |  |  | DG333 |
|  |  | Socket 2 fois 10 pins |
| U3 |  |  | ATtiny85 |
|  |  | Socket 2 fois 4 pins |
| U4 | Pinheader | 4 pins |  |
|  | Pinheader | 4 pins |  |
|  |  |  |  |
| J1 | PinHeaders | 2 Pins | Male |
| J2 | PinHeaders | 2 Pins | Male |
| J3 | Pinheader | 3 Pins | Male |
| Cavalier pour J3 |  |  |
| J4 | Pinheader | 4 pins | Femelle |
| J5 | Pinheader | 4 pins | Femelle |
| J6 | Barrel Jack |  |  |
| J7 | PinHeaders | 2 Pins | Male |
| J8 | PinHeaders | 2 Pins | Male |
|  |  |  |  |
|  | Plaque de montage rapide | SD06N 456 contacts |  |
|  |  |  |  |
|  | Pinheader | 15 pins | Femelle |
|  | Pinheader | 15 pins | Femelle |

Le fichier complet est en lien dans la description de la vidéo

# 8 > Conseils, Retour d’expérience

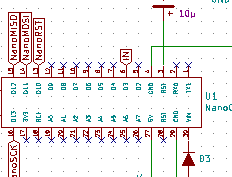
## 8.1. Coupure des pattes de composants traversant :

J’ai remarqué que la forme de cette pince donnait de bons résultats en terme de longueur finie des pattes et de leur régularité :



## 8.2. Axe d’amélioration sur le schéma :

J’ai laissé peut être apparaitre une confusion sur le schéma sur la connexion RST, j’aurais du dénommer D10 autrement que NanoRST à ne surtout pas confondre avec la pinoche 3 RST de l’Arduino Nano.



## 8.3. Nettoyage PCB :

**TRES IMPORTANT :** **Pensez à bien aérer et à vous mettre l’abri de toutes étincelles potentielles et de source de chaleur. LIRE LA NOTICE UN1219 IMPERATIVEMENT.**

On trouve généralement ce type d’alcool dans les drogueries à l’ancienne.

# Annexe 1 – Extrait datasheet DG333



# Annexe 2 – Schéma KiCad



Fichier dans la description de la vidéo